# AN EMBEDDED SYSTEM FOR ENTERTAINMENT ROOM AUTOMATION

OLIVER HOFKENS

NICK VERBAENDERT
TIM DAMS

ABSTRACT

November Five's entertainment room is used for different purposes – hosting presentations, streaming keynotes, taking a break on the PlayStation, to name a few – but the existing setup was annoyingly complex. An embedded system was developed to simplify control of and automate the different devices in the entertainment room: An Onkyo A/V receiver, Epson projector, AppleTV, Telenet Digicorder and WeMo Insight switch.

BEFORE / AFTER

Zone 1 Zone 2 Zone 3
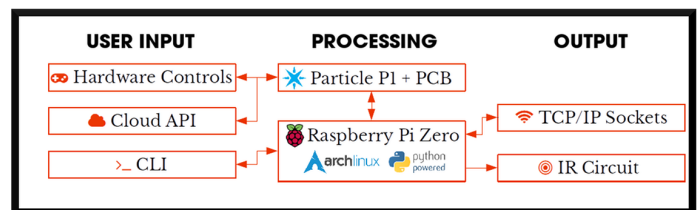
— Analog
— Digital
-- Infrared
-- Network
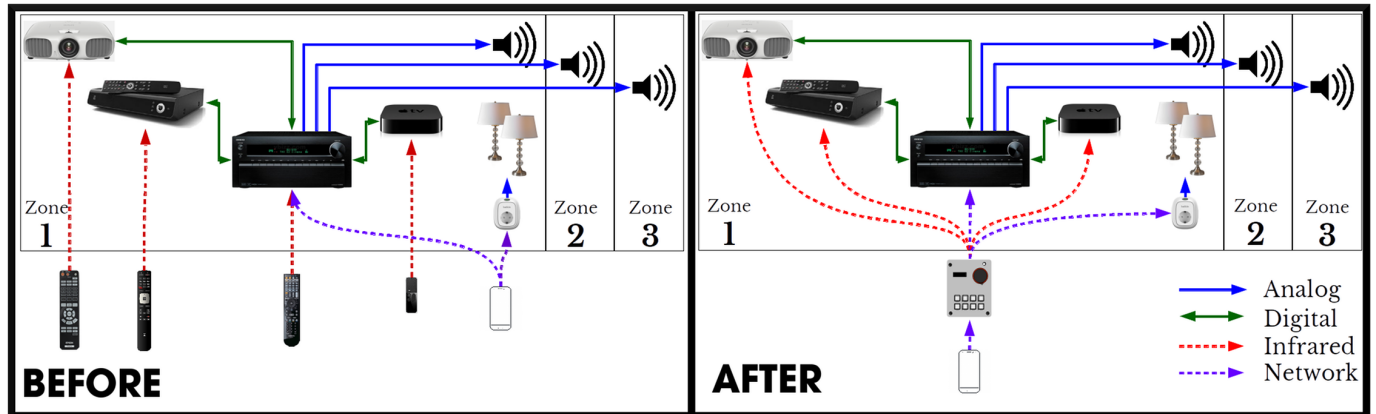
SOFTWARE

The software was written in Python 3 and uses custom libraries based on the 'Ouimeaux', 'Onkyo-eISCP' and 'LIRC' projects to control the WeMo switch, Onkyo receiver and infrared devices respectively. It combines these libraries into a single API by implementing the command and chain-of-responsibility software patterns and organizing the entertainment room as a tree of devices, zones, and settings. To optimize for multithreading-capable platforms and compensate for CPython's global interpreter lock, a communication module was created based on Python's 'multiprocessing' package.

HARDWARE

A Raspberry Pi Zero was chosen to run the Python software on because of its form factor and performance per watt. The rest of the hardware consists of a custom PCB designed with surface-mount technology (SMT), and controlled by a Particle P1. Communication with the entertainment room devices happens through a custom IR-emitting circuit and over WiFi. User-facing hardware includes a rotary encoder with push switch, 8 rubber buttons illuminated by RGB LED's, OLED display, and RGB status LED. A dedicated debugging micro-USB port and reset and mode buttons helped during development. A custom casing was designed and 3D-printed to hold the hardware.

USER INPUT	PROCESSING	OUTPUT
Hardware Controls	Particle P1 + PCB	TCP/IP Sockets
Cloud API	Raspberry Pi Zero	IR Circuit
CLI	archlinux python powered	

FIRMWARE

The firmware for the P1 handles the user-facing hardware and uses Particle's cloud functions to expose a basic REST API and publish device state change events. November Five's Spencer backend consumes this API, so employees can use the Spencer app to check and control all entertainment room devices like all other office domotics.

ARTESIS PLANTIJN HOGESCHOOL ANTWERPEN

NOVEMBER FIVE